Unit 4: Database Backup and Recovery

- Backup and Recovery Overview
- Database backup, restoration and recovery
- defining of backup and recovery procedure
- Testing the backup and recovery plan
- parallel instance recovery
- recovering from non-critical loses

Database must be protected from failures to protect crucial data from losing. There are two major categories of database failures:

- non-media failures: less critical in nature (statement failures, process failures, instance failures, and user errors), and
- media (disk) failures: more critical in nature—the inability to read or write from a database file.

Non Media Failure

- In most cases, statement, process, and instance failures are automatically handled by Oracle and require no DBA intervention. User error can require a manual recovery performed by the DBA.
- *Statement failure consists of a syntax error in the statement, and Oracle usually returns an* error number and description.
- *Process failure occurs when the user program fails for some reason, such as when there is an* abnormal disconnection or a termination. The process monitor (PMON) process usually handles cleaning up the terminated process.
- *Instance failure occurs when the database instance abnormally terminates due to a power* spike or outage. Oracle handles this automatically upon start-up by reading through the current online redo logs and applying the necessary changes back to the database.
- *User error occurs when a table is erroneously dropped or data is erroneously removed.*

Media or Disk Failure

A media failure occurs when the database fails to read or write from a file that it requires. For example, a disk drive could fail, a controller supporting a disk drive could fail, or a database file could be removed, overwritten, or corrupted. Each type of media failure that occurs requires a different method for recovery.

The basic step to perform media recovery are:

1. Determine which files will need to be recovered: data files, control files, and/or redo logs.
2. Determine which type of media recovery is required: complete or incomplete, opened database, or closed database.
3. Restore backups of the required files: data files, control files, and offline redo logs (archived logs) necessary to recover.
4. Apply offline redo logs (archived logs) to the data files.
5. Open the database at the desired point, depending on whether you are performing a complete or an incomplete recovery.
6. Perform frequent testing of the process. Create a test plan of typical failure scenarios.

Backup and Recovery Overview

Backup and **recovery** is one of the most important aspect of database administration.
A backup is a representative copy of data. This copy can include important parts of a database such as the control file, redo logs and datafiles. A backup protects data from application error and acts as safeguard against unexpected data loss, by providing a way to restore original data.
Backups are divided into:

- Physical backups, and
- logical backups

Physical backups are copies of physical database files. The phrase "backup and recovery" usually refers to the transfer of copied files from one location to another, along with the various operations performed on these files.
In contrast, logical backups are those that contains data that is exported using SQL commands and stored in a binary file. Logical backups are used to supplement Oracle server. To recover a restored backup, data is updated using redo records from the transaction log.

Different type of backup strategy may include:

- **The entire database (whole).**
  A whole database backup includes all data files and at least one control file (remember that all control files within a database are identical).
- **A portion of the database (partial).**
  Partial database backups may include zero or more tablespaces, zero or more data files, and may or may not include a control file.

Backup type may be:

- **All information from all data files (full)**
  Full backups make a copy of every data block within the files being backed up that contains data.
- **Only information that has changed since some previous backup (incremental)**

Incremental backups make a copy of all data blocks that have changed since some previous backup.

Backup mode may be:

- **Offline (consistent, cold)**
  Offline backups (also known as consistent backups) are taken while the database is not open. They are consistent because at the time of the backup, the SCN data file headers matches the SCN in the control files.
- **Online (inconsistent, hot)**
  Online backups (also known as hot or inconsistent backups) are taken while the database is open. The backups are inconsistent because with the database open there is no guarantee that the data files are synchronized with the control files. Inconsistent backups require recovery in order to be used.

Backups may be stored as:

- **Image copies:**
  Image copies are duplicates of data or archived log files (similar to simply copying the files using operating system commands).
- **Backup sets:**
  Backup sets are copies of one or more data or archived log files. With backup sets, empty data blocks are not stored, thereby causing backup sets to use less space on disk or tape. Backup sets can be compressed to further reduce the space requirements of the backup.

The advantage of creating a backup as an image copy is improved granularity of the restore operation. With an image copy only the file or files need to be retrieved from tape.
With backup sets the entire backup set must be retrieved from tape before you extract the file or files that are needed.
The advantage of creating backups as backup sets is better space usage. Most databases contain 20% or more empty blocks. Image copies back every single data block up, even if the data block is empty. Backup sets significantly reduce the space required by the backup.

Recovery

A major responsibility of the database administrator is to prepare for the possibility of hardware, software, network, process, or system failure. If such a failure affects the operation of a database system, you must usually recover the database and return to normal operation as quickly as possible. Recovery should protect the database and associated users from unnecessary problems and avoid or reduce the possibility of having to duplicate work manually.
Recovery has two part: rolling forward and rolling back. When Oracle rolls forward, it applies redo records to the corresponding data blocks. Oracle systematically goes through the redo log, which records every changes before the failure of system has occurred, to determine which changes it needs to apply to which blocks and then changes the block. Once Oracle has

completed rolling forward stage, it begins rolling back where Oracle search through the rollback information stored in transaction table for uncommitted transactions, undoing any that it finds.

The Physical data Structures used in recovering data are:

- Datafiles and data blocks
- Control Files
- Rollback Segments
- Redo Log Files

Datafiles and Data Blocks:

A data file is a file which is part of an Oracle database that stores data - including user data and undo data and are collectively known as tablespaces.
Every Oracle database has one or more physical datafiles. The datafile is divided into smaller units called data blocks. Data blocks are the smallest units of storage that the database can use or allocate.
The first block of every datafile is header that contains important information such as file size, block size, tablespace, and creation timestamp. Whenever the database is opened, Oracle checks to see that the datafile header information matches the information stored in the control file. If it doesn't match then recovery is necessary.
Oracle reads data in a datafile during normal operation and stores it in the buffer cache. The Database Writer (DBWR) or DB Writer later writes from buffer cache to disk. The more data that accumulates in memory without being written in disk, the longer the recovery time it will take.

Control files:

The control file is the file that contains the record of the physical structures of the database and their status. Several types of information stored in the control file related to backup and recovery are:

- Database information (RESETLOGS SCN and time stamp)
- Tablespace and datafile records (filenames, datafile checkpoints, read/write status, offline ranges)
- Information about redo threads (current online redo log)
- Log records (log sequence numbers, SCN range in each log)
- A record of past RMAN backups
- Information about corrupt datafile blocks

Every time a user mounts database, its control file is used to identify the datafiles and online redo log files that must be opened for database operation. If physical structure of database changes, a new datafile or redo log file is created. Oracle then modifies the database's control file to reflect the change.
The control file should be backed up whenever the structure of database changes. Loss of the control file makes recovery from a data loss much more difficult.

Rollback Segments (Undo Segments):

Every database contains one or more rollback segments which is the logical structure contained in datafile that records the initial state of information before a transaction modifies a data block.

In general, the rollback segment of a database store the old values of data changed by uncommitted transactions which Oracle can use during database recovery to undo any uncommitted changes applied from the redo log to the datafiles, putting the data into a consistent state.

Redo Log Files:

An Oracle database requires at least two online redo log groups, and in each group there is at least one online redo log member, an individual redo log file where the changes are recorded. Redo logs record all changes made to a database's data files. Each time data is changed in the database, that change is recorded in the online redo log first, before it is applied to the datafiles.

At intervals, the database rotates through the online redo log groups, storing changes in the current online redo log.

Because the redo log contains a record of all changes to the datafiles, if a backup copy of a datafile from some point in time and a complete set of redo logs from that time forward are available, the database can reapply changes recorded in the redo logs, in order to re-construct the datafile contents at any point between the backup time and the end of the last redo log. However, this is only possible if the redo log has been preserved.

Therefore, preserving the redo logs is a major part of most backup strategies. The first level of preserving the redo log is through a process called archiving. The database can copy online redo log groups that are not currently in use to one or more archive locations on disk, where they are collectively called the archived redo log. Individual files are referred to as archived redo log files. After a redo log file is archived, it can be backed up to other locations on disk or on tape, for long term storage and use in future recovery operations.

There are three basic types of recovery:

- instance recovery
- crash recovery
- media recovery

First two are performed by Oracle automatically at instance startup. Latter requires the user to issue command.

Instance recovery:

It occurs in an open database when one instance discovers that another instance has crashed. The surviving instance automatically uses the redo log to recover the committed data in database buffers that was lost when the instance failed.

Crash recovery:

It occurs when either a single-instance database crashes or all instance of multi-instance database crash. In crash recovery, an instance must first open the database and then execute recovery operation.

Media recovery:

It is executed on user's command, usually in response to media failure. In media failure, online or archived redo logs can be used to make a restored backup current or to update it to a specific point in time. Media recovery can restore the whole database, a tablespace, or a datafile and recover them to some non-current time, media recovery is being performed.

Database Backup, restoration and recovery:

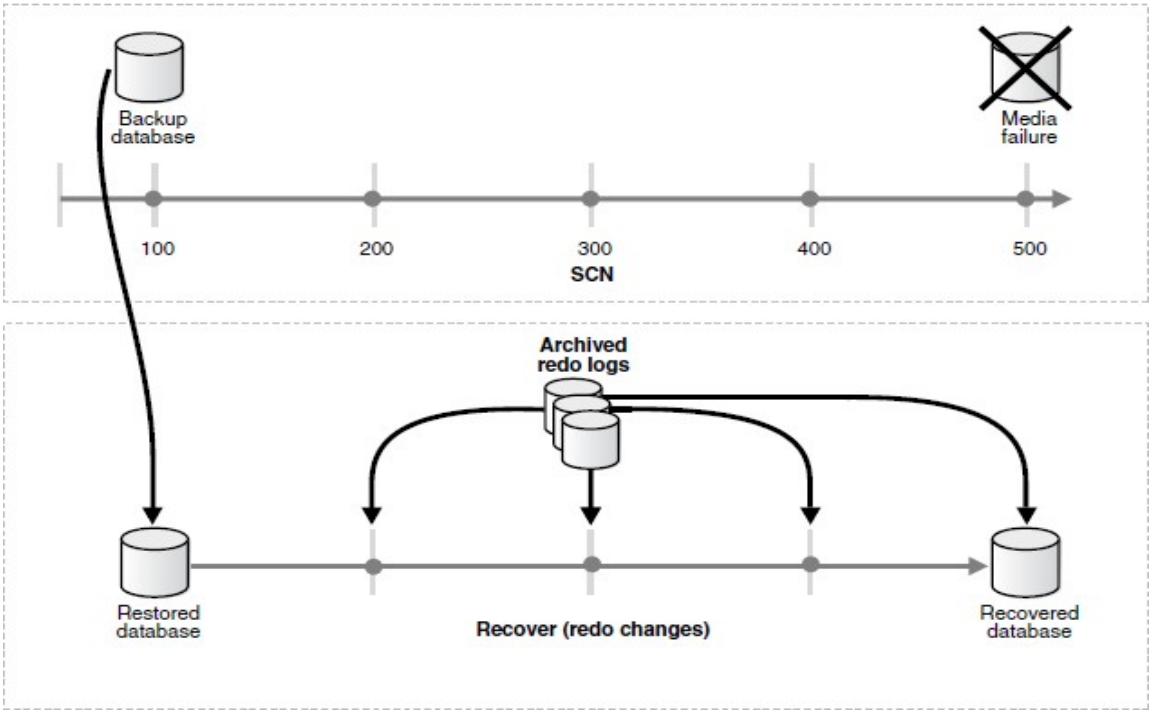
For normal functionality of a database, Oracle consists of set of physical structures that includes datafiles, redo logs, control files, and initialization files. If these files are not present, the database may not start up or it may halt during normal operations. So, all of these files must be backed up on a regular basis to disk, tape, or both. Such a backup can consist of a user- managed backup or Recovery Manager (RMAN)-based backup.

Both of these backups can be used to restore the necessary database files from disk or tape to the desired location.

The restore process consists of copying the database files from tape or disk to a desired location so that database recovery can begin.

The recovery process consists of starting the database and making it consistent using a complete or partial backup copy of some of the physical structures of the database. Recovery has many options depending on the type of backups that are performed.

Figure 1–1 Restoring and Recovering a Database



In this example a full backup of a database (copies of its datafiles and control file) is taken at SCN 100. Redo logs generated during the operation of the database capture all changes that occur between SCN 100 and SCN 500.
Along the way, some logs fill and are archived. At SCN 500, the datafiles of the database are lost due to a media failure.
The database is then returned to its transaction-consistent state at SCN 500, by restoring the datafiles from the backup taken at SCN 100, then applying the transactions captured in the archived and online redo logs and undoing the uncomitted transactions.

Defining a Backup and Recovery Strategy

To create a solid *backup and recovery strategy, you must keep in mind six major* requirements:

1. The amount of data that can be lost in the event of a database failure.
2. The length of time that the business can go without the database in the event of a database failure
3. Whether the database can be offline to perform a backup, and if so, the length of time that it can remain offline.
4. The types of resources available to perform backup and recovery.
5. The procedures for undoing changes to the database, if necessary.
6. The cost of buying and maintaining hardware and performing additional backups versus the cost of replacing or re-creating the data lost in a disaster.

Testing Backup and Recovery Strategies:

One of the most important (but also most overlooked) components of the recovery plan is testing. Testing should be done before and after the database that you are supporting is in production. Testing validates that your backups are working, and gives you the peace of mind that recovery will work when a real disaster occurs.

You should document and practice scenarios of certain types of failures so that you are familiar with them, and you should make sure that the methods to recover from these types of failures are clearly defined. At a minimum, you should document and practice the following types of failures:

- Loss of a system tablespace.
- Loss of a non-system tablespace.
- Loss of a current online redo log.
- Loss of the whole database.

Testing recovery should include recovering your database to another server, such as a test or development server.

Test servers are absolutely necessary, however, and businesses that fail to perform this requirement can be at risk of severe data loss or an unrecoverable situation

.

Defining Backup and Recovery Procedure:

- Datafile Media Recovery: Restore Datafiles, Apply Redo.
- Complete, Incomplete and Point-In-Time Recovery.
- Automatic Recovery After Instance Failure: Crash Recovery.

Datafile Media Recovery: Restore Datafiles, Apply Redo

Datafile media recovery (often simply called media recovery) is the most basic form of user-initiated data recovery that can be used to recover from a lost or damaged current datafile, SPFILE or control file.

Datafile media recovery can be performed whether you use Recovery Manager or user-managed backup and recovery.

The need to restore a datafile from backup is not detected automatically. The first step in performing media recovery is to manually restore the datafile by copying it from a backup. Once a datafile has been restored from backup, however, the database does automatically detect that this datafile is out of date and must undergo media recovery.

Complete, Incomplete/Point-in-time Recovery:

Complete recovery is recovering a database to the most recent point in time, without the loss of any committed transactions. Generally, the term "recovery" refers to complete recovery.

In incomplete recovery, also known as point-in-time recovery, the goal is to restore the database to its state at some previous target SCN or time. Point-in-time recovery is one possible response to a data loss caused by, for instance, a user error or logical corruption that goes unnoticed for some time.

Automatic Recovery after Instance Failure: Crash Recovery

The crash recovery process is a special form of recovery, which happens the first time an Oracle database instance is started after a crash (or SHUTDOWN ABORT).

In crash recovery, the goal is to bring the datafiles to a transaction- consistent state, preserving all committed changes up to the point when the instance failed.

Crash recovery uses only the online redo log files and current online datafiles, as left on disk after the instance failure. Archived logs are never used during crash recovery, and datafiles are never restored from backup.

The database applies any pending updates in the online redo logs to the online datafiles of your database. The result is that, whenever the database is restarted after a crash, the datafiles reflect all committed changes up to the moment when the failure occurred.

(After the database opens, any changes that were part of uncommitted transactions at the time of the crash are rolled back.)

Parallel Instance Recovery:

The goal of the parallel recovery feature is to use computed and I/O parallelism to reduce the elapsed time required to perform crash recovery, single-instance recovery, or media recovery. Parallel recovery is most effective at reducing recovery time when several data files on several disks are being recovered concurrently.

Parallel recovery can speed up both instance recovery and media recovery.

Parallel Recovery using RMAN

For RMAN, the restore and application of incremental backups are parallelized using channel allocation. The RECOVERY_PARALLELISM parameter determines the number of concurrent processes that participates in recovery. Setting the parameter to 0 or 1 invokes serial recovery. With RESTORE and RECOVER statements, Oracle can automatically parallelize all three stages of recovery.

1. **Restoring Datafiles:** When restoring datafiles, the number of channels you allocate in the RMAN recover script effectively sets the parallelism RMAN uses.
2. **Applying Incremental Backups:** When you are applying incremental backups, the number of channels you allocate determines the potential parallelism.
3. **Applying Redo Logs:** Oracle applies redo logs using a specific number of parallel processes as determined by your setting for the RECOVERY_PARALLELISM parameter. The parameter specifies the number of redo application server processes that participates in the instance or media recovery.

During parallel recovery, one process reads the log files sequentially and dispatches redo information to several recovery processes that apply the changes from the log files to the datafiles.

Parallel Instance Recovery

Parallel execution can also improve recovery processing. For the parallel instance recovery, the parallel execution processes must be running when the instance starts up. Use PARALLEL_MIN_SERVERS to define the number of parallel execution servers available for parallel recovery and PARALLEL_MIN_SERVERS to set limit on number of parallel execution processes available for recovery.

Parallel Media Recovery

The PARALLEL clause in the RECOVER DATABASE statement determines the degree of parallelism in media recovery. If you do not give value for RECOVERY_PARALLELISM and if it has some non-zero value, it will be used as a default degree of parallelism for the media recovery.

Parallel recovery using Operating System utilities:

You can parallelize instance and media recovery in two ways by:

- Setting the RECOVERY_PARALLELISM parameter
- Specifying RECOVER statement options

Setting the RECOVERY_PARALLELISM parameter:

The RECOVERY_PARALLELISM parameter specifies the number of redo application server processes participating in instance or media recovery. One process reads log files sequentially and dispatches redo information to several recovery processes that apply the changes from the log files to the datafiles. A value of 0 or 1 indicates that recovery is performed serially by one process and the value cannon exceed the value of PARALLEL_MAX_SERVERS parameter.

Specifying RECOVER statement options:

When you specify RECOVER statement to parallelize instance and media recovery, the allocation of recovery processes to instance is operating system specific. The DEGREE keyword of the PARALLEL clause can either signify the number of processes on each instance of a parallel server or the number of processes to spread across all instances.

Recovering from Non-critical Losses:

Loss of file can be caused by:

- User error
- Application error
- Media failure

There are some files whose loss can be tolerated without going through the restore and recover process, known as "non-critical" loss. A noncritical file loss is one where the database can continue to function.

You fix the problem by taking one of these actions:

- Create a new file.
- Rebuild the file.
- Recover the lost or damaged file.

Damage to a TEMP file:

Datafiles are written by DBWn process and if it is not present or damaged, database will stay in mount state and will not upen. Unlike datafiles, tempfiles are written by server processes servicing the sessions that need some temporary space. So, even if a temp file is not available at startup time, the database will still open. DBWn will however, write a message to the alert

log. The problem will only be apparent when a session require some temporary space.

Restoring a Temporary tablespace:

Temporary tablespaces, and the tempfiles that makes them up, cannot be backed up. The Recovery Manager (RMAN) ignores them completely if you run the REPORT SCHEMA command to show the files that make up the database; the tempfiles will not be listed.
Since you can't back up a tempfile, you cannot restore it. But you can recreate it as follows:

1. Add another tempfile to the damaged temporary tablespace
2. Take the damaged tempfile offline
3. Drop the damaged file Eg:

alter tablespace temp_ts3 add tempfile 'C:\TEMPFILES\TS3- 2.DBF' size 100m;
Tablespace altered.
alter database tempfile 'C:\TEMPFILES\TS3-1.DBF' offline; database altered.
alter database tempfile 'C:\TEMPFILES\TS3-1.DBF' drop; database altered.

An alternative approach would work at the tablespace level than file level as follows:

1. Create a new temporary tablespace
2. Switch your users over the new temporary tablespace via the ALTER DATABASE command
3. Drop the damaged tablespace Eg:

   create temporary tablespace temp_ts4 tempfile 'C:\TEMPFILES\TS4-1.DBF' size 100m;
   tablespace created.
   alter database default temporary tablespace temp_ts4; database altered.
   drop tablespace temp_ts3 including contents and datafiles;

Tablespace dropped.

Damage to an Online Redo Log File Member

Online redo log is the guarantee that the database will never be corrupted, provided that there is atleast one valid member of the current group (and of any previous groups that is still ACTIVE). If the redo log files are multiplexed, damage to an individual member is not critical but there is a risky position to be in. If there is still a valid member in group, the instance will remain open, and your users will not be aware of any problem—but there will be indications that something is wrong. First, the view V$LOGFILE will show the damaged or missing member as being INVALID. Second, the background processes will report the problem.

Re-creating a Damaged online redo log file member

You have two options: either drop the damaged member and add a replacement number, or clear the group.
In the first option, the damaged member is dropped and then the physical file is deleted from disk. Then a new member is added. This will fix the problem.
In second approach, we use the CLEAR LOGFILE command, which will create a new group by replacing all members and reusing the old member's filenames.
Choice between DROP/ADD and CLEAR depends on circumstances.
If there is no damage to disks and only to the files, CLEAR is simpler to use. Oracle will not let

you CLEAR logfilegroup if that needs to be archived, or if one is still current of active. If there is damage to the disk, CLEAR will fail because Oracle will not be able to re-create the file in its original location. This is when one must DROP the original member and then ADD a replacement member in a different disk location.

Damage to Index Tablespace

Index data is real data that must be protected. Any transaction affecting a table will also affect the index on the table. If an index is not available because of media damage, end users will receive a message which is generated whenever any insert or delete operation is attempted against a table with an index in the damaged tablespaces. Apart from errors being reported to user sessions, damage to an index tablespace datafile will also show up through the usual message in the alert log and backgroundtrace files, and entries in the dynamic performance view.

Recovering an Index Tablespace:

An index tablespace can be restored and recovered like any other tablespace, provided that you are running database in archivelog mode. An alternative method rather than repairing damage through restore and recovery imply the dropping of tablespace, re-creating it and regenerating indexes.
The routine is as follows:

1. Take the damaged tablespace offline
2. Determine which indexes were in the damaged tablespace
3. Drop the tablespace and delete its datafiles
4. Create a new tablespace
5. Generate all the indexes in it To see indexes in tablespace:

SQL> select owner, segment_name from dba_segments where tablespace_name='INDX' and segment_type='INDEX';

To drop tablespace:

SQL> drop tablespace indx including contents and datafiles; To create a new tablespace:
SQL> create tablespace indx datafile 'C:\DATAFILES\INDX-1.DBF' size 100m extent management local uniform size 128k segment space management autonologging;

Damage to the Password file:

The password file is one of the files that cannot be backed up by RMAN. Loss of password file is not critical. It is always possible to connect to a database with the SYSDBA privilege with operating authentication. If the password file is damaged, the database will continue to work normally, except that the remote SYSDBA connections will not be possible. But if the database is shut down, then startup will fail when the instance tries to read the password file. To get around this problem, set the REMOTE_LOGIN_PASSWORDFILE instance parameter to NONE. This will prevent Oracle from looking for a passwordfile and the instance will open normally.
SQL> alter system set remote_login_passwordfile=none scope=spfile; and then restart.

Replacing the password file:

It is possible to backup the password file with normal operating system backup procedures. The system administrators' regular backup of ORACLE_HOME directory will have copies of it and so you can restore it with no problem. An alternative is to re-create it by re-running the command used to create it in the first place as.

orapwd file=<filename> password=<password> entries=<max_users>

Having re-created the password file, reset the REMOTE_LOGIN_PASSWORDFILE parameter to use it.

BACKUP PRINCIPLES

- **Physical and Logical Backups**
- **Whole database and partial database Backups**
- **Consistent and Inconsistent Backups**
- **Offline and Online Backups**
- **RMAN and USER-Managed Backups**

Physical Backups	Logical Backups
Physical backups are backups of physical database files: datafiles and control files.	Logical backups are exports of schema objects into a binary file
Physical backups are divided into two categories: image copies and backups in a proprietary format.	Logical backups have two utilities: Import and Export to move Oracle data in and out of system file.
Image copy is an exact duplicate of a datafile, control file or archived log using COPY command and restore using RESTORE command.	Import is the utility that reads export files and restores the corresponding data into an existing database.
The BACKUP command generates a backup set which is a logical object containing one or more backup pieces where each backup piece is a physical file in a proprietary binary format.	Export writes data from an Oracle database to binary operating system files. These export files store information about schema object eg: tables and stored procedures

Whole Database Backups	Partial Database Backups
Whole database backup includes backups of the current control file along with all datafiles.	A partial database backup is a one which takes backup of some specific files while the database is open or shut down. Such as: Tablespace Backups, Datafile Backups, Control File Backups

Whole database backup must be done when backup is carried out for the first time.	Partial database backup is done only after whole database backup has been done once.
Whole database backup takes backup of each and every files in the database.	Partial database backup can be either incremental or differential which will take backup of those files where changes has been recorded.

Consistent Backup	**Inconsistent Backup**
It is a backup of one or more database file that you make after the database has been closed with a clean SHUTDOWN command.	It is a backup of one or more while the database is open or after the database has shut down abnormally.
In this backup, read/write datafiles and control files are checkpointed with respect to the same **System Change Number (SCN).**	In this backup, all read/write datafiles and control files are not checkpointed with respect to same SCN.
A consistent backup is one in which database has been shut down. It is also called **cold backup**.	An inconsistent backup is taken when the database is up and running. This is also called a **hot backup.**
The database is guaranteed to be in a consistent state because no one can possibly be using the database; it is down.	While the backup is being performed, the database can still undergo change. A datafile in your backup can have old transactions and new transactions. This means the datafile is not consistent with a single point in time.

Offline Backup	**Online Backup**
Offline backup is a backup of the database while it is not running.	An online backup allows you to backup the database while users are working.
to perform our backup we will shutdown the database from RMAN and then mount the database.	to perform online backup you will need to put your database in ARCHIVELOG mode
Also known as cold backup	also known as hot backup

Process to take backup: RMAN>shutdown immediate RMAN>startup mount RMAN>backup database; RMAN>sql? alter database open?;	RMAN>backup database plus archivelog delete input; This command will backup your database. Along with the database backup, it will backup all the archived redo logs that have been generated by your database
While the database is “offline,” the following files should be backed up: All datafiles. • All controlfiles. • All archived redo log files. The init.ora file or server parameter file (SPFILE).	While the database is open, the following files can be backed up: • All datafiles • All archived redo log files One control file, via the **alter database backup controlfile** The server parameter file (SPFILE).